

[enabling a programming language editor having a character position cursor and a randomly positionable pointer;

partially compiling available ones of a plurality of programming language statements in said computer program;

defining] generating a [finite] set of programming language statement information [that is] relevant to [at least one segment of a] modifying the present programming language statement; [from among said plurality of programming language statements that is proximate to said character position; and]

[generating a passive assist window that contains said finite set of programming language statement information in a location proximate to said character position cursor]

receiving a representation of a selection by the computer programmer from the generated set of programming language statement information; and

modifying the present programming language statement based at least in part on the selected programming language statement information.

2. (Amended) [A method according to] The computer-readable medium of claim 1, [including:] wherein the generating step is automatically performed in response to a change in the current program statement [automatically attempting said steps of claim 1 for each character received by said programming language editor].

3. (Amended) [A method according to] The computer-readable medium of claim [2] 1, [including: attempting said steps of claim 1 on a randomly selected one

Q³
of said plurality of programming language statements] wherein the generating step is performed in response to a real-time request by [said] the computer programmer.

5. (Amended) [A method according to] The computer-readable medium of claim [1] 21, wherein the displayed one or more passive assist windows include at least [including: generating a simultaneous plurality of passive assist windows that each contain a finite set of programming language statement information in a location proximate to said character position cursor, said simultaneous plurality of passive assist windows being selected from at least one of a group comprised of:] a selection menu assist window [and] or an informational display assist window.

Q⁴
6. (Amended) [A method according to] The computer-readable medium of claim 1, wherein [said] the generating step [of defining] includes[: generating said] displaying [finite list as] a selectable list of [menu] items [that can each validly complete said at least one segment of said] for modifying the present programming language statement [that is proximate to said character position cursor].

7. (Amended) [A method according to] The computer-readable medium of claim [1] 6, wherein [said step of generating includes: creating] the selectable list of items are displayed in a selection menu assist window [comprised of said list of menu items]; and having further computer-executable instructions for enabling window control features for [said] the selection menu assist window.

8. (Amended) [A method according to] The computer-readable medium of claim 6, [including:] having further computer-executable instructions for replacing [said at least one segment of said] at least a portion of the present programming language statement with one of [said] the list [of menu] items in response to an input command by [said] the computer programmer.

9. (Amended) [A method according to] The computer-readable medium of claim 1, [wherein said] the step of [defining] generating the set of programming language statement information includes having further computer-readable instructions for:

*a*⁴
generating an argument list [of each argument in said] for the present programming language statement; and

identifying an argument type for [each] at least one argument in [said] the argument list selected from at least one of a group comprised of: a mandatory argument and an optional argument.

10. (Amended) [A method according to] The computer-readable medium of claim 9, [wherein said step of generating includes] having further computer-executable instructions for:

reverse parsing [said] the present programming language statement into a plurality of tokens that each represent an individual component selected from at least one of a group comprised of: an object entity segment and a delimiter, in response to a real time request by [said] the computer programmer;

distinguishing [said] the plurality of tokens between a procedure call token and any argument token in [said] the argument list; and

binding [said] the argument list.

11. (Amended) [A method according to] The computer-readable medium of claim 1, [wherein said step of generating includes] having further computer-executable instructions for:

generating an informational display assist window based on an argument list;

distinguishing a mandatory argument from an optional argument within [said] the argument list; and

highlighting a present argument within [said] the argument list that corresponds to a present location [of said character position cursor] within [said] the present programming language statement.

12. (Amended) A system for passively assisting a user in real-time to [complete] modify a programming language statement, [said] the system comprising:

a programming language editor having a character position cursor and a randomly positionable pointer;

means for partially compiling available ones of a plurality of programming language statements in [said] the computer program; [and]

means for generating an assist window that contains a [finite] set of programming language statement information in a location proximate to [said] the character position cursor, [said] the assist window being selected from at

least one of a group comprised of: a selection menu assist window and an informational display assist window;

means for receiving a selection by the user from the set of programming language statement information; and

means for modifying a present programming language statement based at least in part on the selected programming language statement information.

13. (Amended) [A] The system [according to] of claim 12, wherein [said] the means for generating includes:

means for identifying a desired menu item from [said] the selection menu assist window; and

means for replacing a segment of [a] the present programming language statement at a present location of [said] the character position cursor with [said] the desired menu item in response to [said] the means for identifying.

14. (Amended) [A] The system [according to] of claim 12, including: means for displaying information in an informational display assist window, [said] the information being related to at least one segment of [a] the present programming language statement [that is proximate a present location of said character position cursor] and [selected] from at least one type of a group comprised of: a symbol definition, a defined constant, a procedure call map, and an enumerated list.

15. (Amended) [A] The system [according to] of claim 12, wherein the contents of the assist window are automatically updated in response to a change in the current program statement [including: means for automatically

enabling said means of claim 12 for each character received by said programming language editor].

16. (Amended) [A] The system [according to] of claim 12, including: means for [enabling said means of claim 1] specifying the present programming language statement and means for generating the assist window on a [randomly] selected one of [said] the plurality of programming language statements in response to a [real time] request by [said] the user[and independent of any automatic assist feature].

17. (Amended) A real-time method for assisting a user to [complete] modify a programming language statement in a computer program, [said] the real-time method comprising:

enabling a programming language editor having a character position cursor;

continuously resolving symbolic portions of available ones of a plurality of programming language statements into a partial program compilation;

identifying a present programming language statement and at least one segment of [said] the present programming language statement based on a location of [said] the character position cursor;

determining a finite set of information related to [said] the present programming language statement and [said] the at least one segment of [said] the present programming language statement based on [said] the partial program compilation; [and]

generating an assist window of [said] the finite set of information;

receiving a representation of a selection by the
computer programmer from the finite set of information; and
modifying the present programming language statement to
based at least in part on the selected information.

18. (Amended) [A] The method [according to] of
claim 17, [wherein said step of identifying includes]
including the steps of:

determining an identity of input to [said] the
programming language editor by [said] the user;

enabling a reverse parse evaluation of [said] the
present programming language statement into identifiable
tokens for each of [said] the at least one segment therein
in response to [said] the input being an on-demand request
by [said] the user;

enabling execution of a editing task in response to
[said] the input being a programming language editor
command;

enabling a first type of commit of an identified menu
item from a selection menu assist window in response to
[said] the input being a commit key, wherein [said] the step
of enabling a first type of commit includes:

identifying [said] the commit key as a non-delimiter
type commit key; and

discarding [said] the commit key;

enabling a second type of commit of an identified menu
item from a selection menu assist window in response to
[said] the input being a commit key, wherein [said] the
second type of commit includes:

identifying [said] the commit key as a delimiter type commit key; and

inserting [said] the commit key after [said] the identified menu item in [said] the present programming language statement; and

adding to [said] the present programming language statement at a location of [said] the character position cursor in response to [said] the input being a non-commit key type input character.

19. (Amended) [A] The method [according to] of claim 17, wherein [said] the step of generating includes:

displaying a selection menu assist window where [said] the present programming language statement is identified as an operator embedded programming language statement; and

displaying an informational display assist window where [said] the present programming language statement is identified as a non-operator embedded programming language statement.

20. (Amended) [A] The method [according to] of claim 19, wherein [said] the non-operator embedded programming language statement is a procedure call.

Add new claims 21-49.

21. The computer-readable medium of claim 1, having further computer-executable instructions for displaying one or more passive assist windows indicating at least a portion of the generated programming language statement information.

22. The computer-readable medium of claim 1, wherein the programming language statement is modified in response

to the receipt of an indication that a commit key has been activated.

23. The computer-readable medium of claim 1, wherein the representation of the selection is received in response to the activation of a commit key.

24. The system of claim 12, including means for receiving an indication that a commit key has been activated; and wherein the programming language statement is modified in response to the activation of the commit key.

Q5 25. The system of claim 12, wherein the representation of the selection is received in response to the activation of a commit key.

Sub E3 26. The computer-readable medium of claim 17, wherein the programming language statement is modified in response to the receipt of an indication that a commit key has been activated.

27. A computer-readable medium having computer-executable instructions for performing the steps recited in claim 26.

Sub E4 28. The computer-readable medium of claim 17, wherein the representation of the selection is received in response to the activation of a commit key.

29. A computer-readable medium having computer-executable instructions for performing the steps recited in claim 28.

4
30. A computer-readable medium having
computer-executable instructions for performing the steps
recited in claim 17.

31. A computer-readable medium having
computer-executable instructions for performing the steps
recited in claim 18.

5
32. A computer-readable medium having
computer-executable instructions for performing the steps
recited in claim 19.

33. A method in a computer system for supplementing an
incomplete computer programming statement, the method
comprising the steps of:

(a) displaying the incomplete computer programming
statement;

(b) proximate to the display of the incomplete
computer programming statement, displaying a list of one or
more textual programmatic entities;

(c) receiving a user input selection of one of the
displayed textual programmatic entities; and

(d) adding the selected textual programmatic entity to
the displayed statement.

34. The method of claim 33, further comprising the
step of:

(e) receiving user input entering the incomplete
programming statement,

and wherein step (b) is performed in response to
step (e).

35. The method of claim 33, further comprising the step of:

(e) receiving user input specifying a supplement command,

and wherein step (b) is performed in response to step (e).

36. The method of claim 33 wherein step (b) displays textual programmatic entities that can validly be included in the partial statement.

Q5 37. The method of claim 33 wherein the partial statement includes an object identifier, and step (b) displays members of an object identified by an object identifier.

38. The method of claim 33 wherein the partial statement includes a portion of a symbol, and wherein step (b) displays complete symbols in which the portion of a symbol is contained.

39. The method of claim 33 wherein the partial statement is a partial assignment statement identifying a variable to which a value is to be assigned, and wherein the identified variable has a type, and wherein the type has possible values, and wherein step (b) displays possible values of the type.

40. The method of claim 33 wherein the partial statement is a partial function call statement identifying a function, and wherein the identified function has one or more parameters, and wherein step (b) displays parameters of the identified function.

41. A computer-readable medium whose contents cause a computer system to supplement a computer programming statement by performing the steps of:

- (a) displaying the computer programming statement;
- (b) proximate to the display of the incomplete computer programming statement, displaying one or more textual programmatic entities;
- (c) receiving a user input selection of one of the displayed textual programmatic entities; and
- (d) adding the selected textual programmatic entity to the displayed statement.

as 42. The computer-readable medium of claim 41 wherein the contents of the computer-readable medium further cause the computer system to perform the step of:

(e) receiving user input entering the programming statement,

and wherein step (b) is performed in response to step (e).

43. The computer-readable medium of claim 41 wherein the contents of the computer-readable medium further cause the computer system to perform the step of:

(e) receiving user input specifying a supplement command,

and wherein step (b) is performed in response to step (e).

44. The computer-readable medium of claim 41 wherein the contents of the computer-readable medium further cause the computer system to perform the step of:

(e) displaying in conjunction with the statement a position indicator indicating a position within the incomplete statement,

and wherein step (b) displays textual programmatic entities that can validly be included in the statement at the position indicated by the position indicator.

45. The computer-readable medium of claim 41 wherein the statement includes an object identifier, and wherein step (b) displays members of an object identified by the object identifier.

46. A computer-readable medium of claim 41 wherein the statement includes a portion of a symbol, and wherein step (b) displays complete symbols in which the portion of a symbol is contained.

47. The computer-readable medium of claim 41 wherein the statement is an assignment statement identifying a variable to which a value is to be assigned, and wherein the identified variable has a type, and wherein the type has possible values, and wherein the step (b) displays possible values of the type.

48. The computer-readable medium of claim 41 wherein the statement is a function call statement identifying a function, and wherein the identified function has one or more parameters, and wherein step (b) displays parameters of the identified function.

49. An apparatus for supplementing an incomplete computer programming statement, comprising:

a display device displaying an incomplete computer programming statement, the display device further